Remarks/Arguments

This Amendment is in response to the Final Office Action mailed 06 April 2004 (06.04.2004). In this Final Office Action, the Examiner maintained his rejection of claims 1-4 and 7 under 35 USC 103(a) as being unpatentable over Cottle et al. (US Pat. No. 6,263,396) in view of Applicant's Admitted Prior Art. For the same reasons, the Examiner also rejected claims 9-12, 14-17, and 19-22. The Examiner found allowable subject matter in claims 5, 13, 18, and 23, if rewritten in independent form including all of the limitations of the base claim and any intervening claims.

1. Summary of Current Claims

Claims 2-5, 7, and 9-23 remain in this application. Claim 1 has been canceled in this amendment, and claims 6 and 8 were previously canceled. Claims 2-5 have been amended to depend from independent claim 7. Independent claims 9, 14, and 19 have been amended to clarify that the claimed "synthesizable microprocessor core" is specifically an ARM-type core and to eliminate references to "first" and "second" program execution streams in the claimed processor core element, the interrupt controller element, and the interrupt service routine element and specify ARM and THUMB, as appropriate.

2. Claim Rejections under 35 USC 103(a)

The Examiner maintained his rejection of claims 1-4 and 7 under 35 USC 103(a) as being unpatentable over Cottle et al. (US Pat. No. 6,263,396) in view of Applicant's Admitted Prior Art, and rejected claims 9-12, 14-17, and 19-22 for the same reasons. In response to the Office Action and Applicant's telephone conference with the Examiner

on 28 June 2004, Applicant is canceling claim 1, amending claims 2-5 to depend from claim 7, and amending independent claims 9, 14, and 19 clarify that

> --the processor core is an ARM core and that selectively processes ARM and THUMB program execution streams, where the THUMB program execution stream is more economical with program code space than the ARM execution stream, and where the core switches to ARM execution upon receiving an interrupt request;
>
> --the interrupt controller receives interrupt requests associated with interrupt service routines programmed in THUMB code; and
>
> --the shared interrupt service routine preamble is coded in ARM code to switch the core to THUMB program execution before executing the THUMB-encoded interrupt service routines.

Previously, claims 1, 9, 14, and 19 used more generic language (i.e., "first" and "second" program execution streams) to describe the ARM and THUMB program execution modes, their relationship to each other and the processor core, and to specify which was used in the claimed interrupt service routine preamble. Only claim 7, an independent apparatus claim with no dependent claims, specifically limited the claim elements to an ARM core and ARM and THUMB program execution streams.

The Examiner initially rejected the claims, contending that Cottle teaches an interrupt service routine preamble shared amongst a plurality of interrupt service routines, and citing various sections of Cottle's disclosure. Applicant responded that, while Cottle is obviously aware of the ARM and THUMB program execution modes of the ARM core, Cottle's invention neither focuses on nor exploits the two different program modes for processing interrupts. Cottle teaches the use of a programmable interrupt mask and a programmable interrupt vector table to enable the masking of

interrupts, the prioritization of interrupts, and the servicing of multiple interrupts. Applicant provided further explanation of the instant invention, and then explained in detail how the sections of Cottle's disclosure cited by the Examiner do not, in fact, teach the claimed shared interrupt service routine preamble encoded in ARM code that switches the ARM core back to THUMB execution before executing the appropriate interrupt service routine. Rather than repeating that explanation, Applicant's response to the First Office Action, filed on 14 January 2004, is incorporated by reference in its entirety.

In response, the Examiner cited col. 65, line 59--col. 68, line 47 as providing additional support for the same rationale for rejection. The Examiner stated that this section teaches the various claim elements and limitations, and in particular, "an interrupt service routine is coded in the first program execution stream [presumably, ARM] to cause the hardware switch to the second program execution stream [presumably, THUMB]." Final Office Action, p. 5. First, Applicant disputes that these lines teach the interrupt service routine described by the Examiner. However, more importantly, the present invention as described and claimed does not include "an interrupt service routine...coded in the first program execution stream [presumably, ARM] to cause the hardware switch to the second program execution stream [presumably, THUMB]." As described in the instant application, the present invention is a shared interrupt service routine **preamble** that *avoids* the use of interrupt service routines that include ARM instructions, because ARM code requires too much program space. Application p. 2, lines 19-27; p. 8, line 20-35. The present invention uses a single, shared interrupt service routine preamble encoded in ARM instructions to put the

processor into THUMB execution mode. This allows programmers to avoid using *any* ARM code in interrupt service routines, because after the execution of the interrupt service routine preamble, the core is in the THUMB execution mode and interrupt service routines can be encoded in THUMB instructions.

In Col. 65, line 59--col. 68, line 47 Cottle teaches us about the ARM processor's exception vectors in general, and about two exception vectors—associated with the reset and undefined instruction vectors—in particular. In Table 26, Cottle lists the various types of exceptions that interrupt normal program flow, and in general, how the ARM core handles those exceptions. Cottle's use of the word "mode" in this table should not be confused with the use of the word "mode" in the context of "ARM mode" or "THUMB mode." "ARM mode" and "THUMB mode" refer to the addressing capability--32-bit or 16-bit–that the ARM processor uses in fetching, loading, and processing instructions and data, regardless of whether those instructions or data belong to a normal application program, an interrupt service routine preamble, or any other instruction encountered during operation. On the other hand, the "modes" listed in Cottle's Table 26, including supervisor mode (SVC), undefined mode, abort mode, IRQ mode, and FIQ mode, refer to the processor's current operating "rules" relating to degree of memory access, access to certain registers, and ability to manipulate the program counter. The ARM processor's normal operating mode is the "user" mode, which allows normal program execution and access to all 16 general purpose registers. All other modes (supervisor, undefined, abort, system, FIQ, and IRQ) are "privileged modes" which are entered to service exceptions and interrupts, or to access protected resources. For example, in FIQ mode, five of the sixteen general purpose registers are

reserved for the use of the fast interrupt handler. In Table 26, Cottle is simply referring to the operating mode the processor enters when it encounters the listed exception or interrupt.

In the text associated with Table 26, Cottle does refer to the switch from ARM to THUMB and vice versa, in col. 66, lines 42-45. Here, Cottle explains that ARM generates an exception when it encounters an unrecognized instruction, such as might occur if a mistake is made when transitioning between the ARM and THUMB mode, or if a prefetched instruction is overwritten.

Clearly, then, Col. 65, line 59-Col. 66, line 47 does not teach the use of a shared ARM-encoded interrupt service routine preamble that switches the ARM core into THUMB mode to execute a THUMB-encoded interrupt service routine, and the newly-cited section does not support the Examiner's continuing claim rejections under §103(a).

In response to the First Office Action, Applicant argued that the Examiner's §103 rejection was improper because, among other things, Cottle neither disclosed nor suggested the use of an ARM-encoded interrupt service routine preamble that switches the ARM core to THUMB mode and that is shared amongst interrupt service routines. Therefore, a prima facie case of obviousness, using Cottle and the BX instruction as 103(a) references, cannot be established. The Examiner responded that this feature was not recited in the rejected claims, listing claims 1, 9, 14, and 19. In a conversation with the Examiner on June 28, 2004, Applicant pointed out that these features were, in fact, recited in rejected claim 7, and the Examiner conceded the error. Applicant has canceled claim 1, which was an apparatus claim amended claims 2-5 to

depend from claim 7. In addition, Applicant has amended claims 9 (a system claim), and 14 and 19 (method claims) to conform in substance to claim 7. Accordingly, after the current amendments are entered, all of the pending claims to either expressly recite a shared ARM-encoded interrupt service routine preamble that causes a hardware switch to THUMB program execution, or depend from a claim that expressly recites a shared ARM-encoded interrupt service routine preamble that causes a hardware switch to THUMB program execution. Given the above discussion relating to the newly-cited Col. 65, line 59-Col. 66, line 47, and the current amendments, Applicant respectfully requests that the Examiner withdraw the rejections to claims 1-5, 7, and 9-23 under 35 USC 103(a) as being unpatentable over Cottle et al. (US Pat. No. 6,263,396) in view of Applicant's Admitted Prior Art.

3. Summary

In view of the above, Applicant believes that each of the presently pending claims is in immediate condition for allowance or appeal. Accordingly, Applicant respectfully requests that the Examiner enter the above amendments, withdraw the outstanding objections and rejections of the claims, and issue a timely Notice of Allowance in this case.

Respectfully submitted,

Karen S. Wright

Date: August 5, 2004

Karen S. Wright
Reg. No. 45,240

Matthew J. Booth
Karen S. Wright
Booth & Wright, L.L.P.
PO Box 50010
Austin, Texas 78763-0010
Telephone: (512) 474-8488
Facsimile: (512) 474-7996
matthew.booth@boothlaw.com
karen.wright@boothlaw.com
http://www.boothlaw.com/